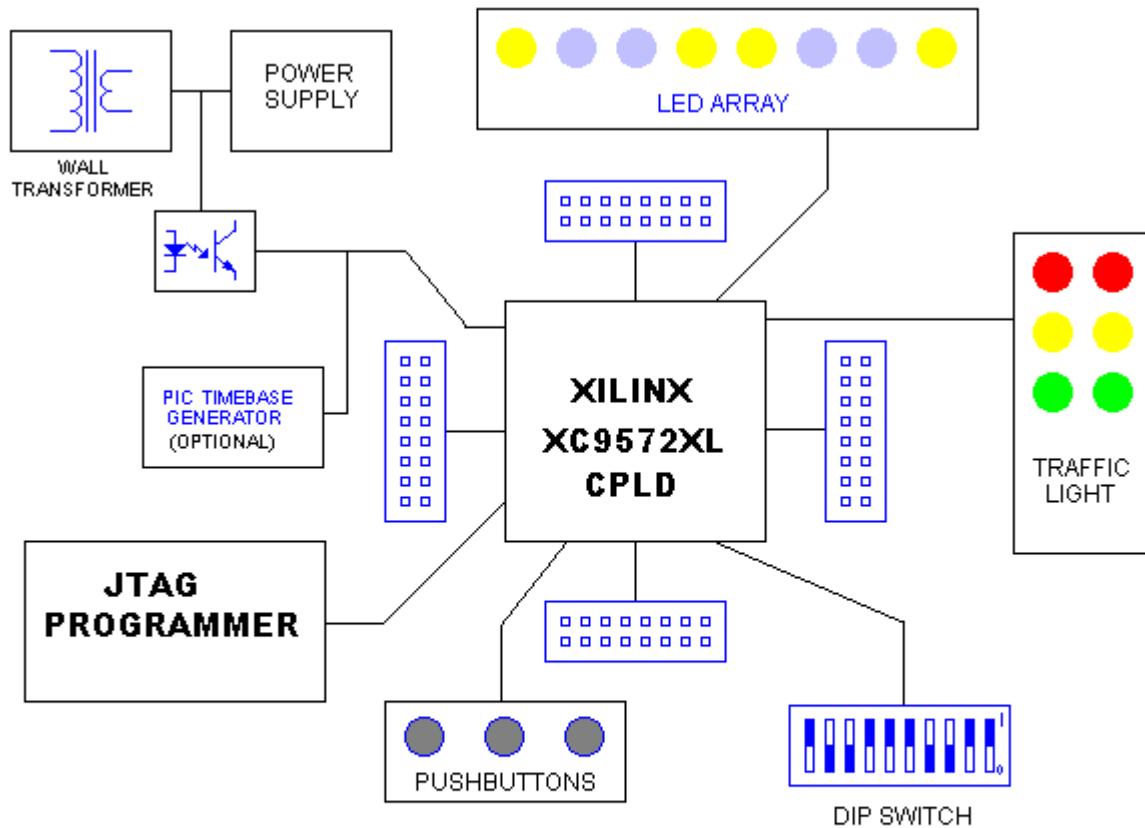


Lab9500™ by EZ-Ware

The low-cost Lab9500board provides all the hardware necessary for the laboratory portion of an entry-level course that covers the basics of digital design. A Xilinx XC9572XL CPLD (Complex Programmable Logic Device) forms the heart--or the brains--of the board. The Xilinx XC9500 family provides macrocells based on PAL-style registered logic, and the family capabilities range from the XC9536 with 36 macrocells to the XC95288 with 288 macrocells. (See, "What's a Macrocell?") A block diagram of the Lab9500 follows.



LAB9500 BLOCK DIAGRAM

The Lab9500board supplies the following I/O devices for student use:

- Eight single-color LEDs,
- Three pairs of colored LEDs, arranged as 2 red, 2 amber, and 2 green (simulates a traffic light at a cross street),
- Ten on/off DIP switches, and
- Three momentary pushbuttons, (two debounced).

The ten DIP switches makes it easy to input data. Two debounced pushbuttons provide clean strobe signals that let the CPLD "sense" individual external events. The 14 LEDs let students see the logic state at each of 14 logic outputs on the circuits they devise.

For timing, an optical isolator extracts a 60-Hz signal from an AC wall transformer that powers the board. A socket is provided for an optional timebase generator made with a PIC microcomputer. In addition, the board routes all of the CPLD chip's I/O lines to header pads. By mounting suitable headers on the board, students can connect other logic devices or a custom I/O board to the Lab9500board.

The Lab9500board uses a XC9572XL CPLD in a 64-pin flatpack (500 micron centers). While all macrocells are not available at I/O pins, many CPLD applications typically use registers, counters and other intermediate signals that do not go to I/O pins.

What's the catch? Usually, programmable devices need an expensive programmer and/or expensive software. The software needed to generate designs for the 9500 family and to physically program them into the chip is provided free of charge from Xilinx. This does require a programmer that uses a JTAG interface. The Lab9500, however, has a built-in JTAG programmer interface. All that is needed is a DB25 M/F cable to plug into the printer port of any PC. (The cable is included with the Lab9500).

PLD Background

Electronic designers have used programmable logic devices (PLDs) for over 25 years. Two things, however, have prevented them from becoming common elements in simple digital designs. First, the early PLDs were expensive--\$5 to \$10 each--and they required a sophisticated electronic "programmer" to "burn" a logic configuration in them. Second, the original PLDs used a fusible-link technology. Once the programmer burned a fuse, no way existed to return it to its original state. So, a chip could serve in only one application. And, if an engineer discovered a "bug" in the programmed information, he or she had to discard the chip and program the corrected information in a new device. In those early days, designers could justify using PLDs only in high-volume applications that could absorb high development costs.

A major development occurred when Altera (San Jose, CA) first used metal-oxide semiconductor (MOS) technology to produce a PLD. This technology produced a structure that users could program as needed, but that returned to its original, unprogrammed state upon exposure to intense ultraviolet light. (The PLD packages included a quartz-glass window that admitted the UV light to the internal chip.) Although the MOS chips were still expensive, designers could erase them and use them again and again. Later, Lattice (Hillsboro, OR) and other companies implemented PLDs that took advantage of electrically erasable semiconductor devices. In these devices, electrical signals erased information, so the chips no longer needed quartz windows. The new, inexpensive packages cost less than \$5, or close to \$2 in large quantities. These PLDs still needed a special programmer, though.

Recently, manufacturers of both PLDs and microcontrollers have adopted programming algorithms and techniques that use a standard serial interface. This interface has simplified programming and eliminated the need for a separate, expensive device programmer. Now, a PC and a simple, inexpensive circuit serve to program PLDs and microcontrollers. PLDs in the Xilinx XC9500 family take advantage of a special serial standard called JTAG. The JTAG standard was developed to let test engineers check the internal operations of digital ICs. Eventually it evolved so that people can use it to program the internal operations of chips, including PLDs. The JTAG port, also called a boundary-scan port (IEEE 1149) now comes as standard on many ICs. Xilinx provides low-cost interface hardware and software that uses a PC's printer connection to simulate a complete JTAG port.

PLDs found in a single-use product such as a home thermostat or an automotive engine controller do not need reprogramming. Thus, the PLDs found in these applications need no additional programming circuitry on board. The CPLD on the Lab9500board, however, gets programmed again and again, based on the experiments students have underway. The Lab9500board, therefore, comes with a built-in JTAG interface. The only additional hardware required for programming is a low-cost cable (provided) that connects the Lab9500board to a PC's printer port.

Experiments with Combinational Logic

The macrocell found in the CPLD chip on the Lab9500board furnishes an AND/OR architecture that can implement virtually any combinational circuit or function. The list below shows some of the logical functions a student could implement and test using the board:

Eight-input AND, NAND, OR, or NOR gate,
Four-bit full adder,
Quad 2:1 multiplexer,
Three-to-eight decoder, high- or low-true output,
Eight-to-three priority encoder, and
Hex seven-segment decoder/driver

The functions above show students how to use high-level software for circuit design with a CPLD and let students investigate how they can configure a CPLD to operate as standard logic blocks.

Experiments with Clocked Logic

Many circuits require clocked logic that progresses through a sequence of operations based on an input clock signal. The D- and T-type flip-flop circuits provide the basis for a range of clocked-logic experiments. The following list details some of the basic clocked-logic elements and some larger functions built by combining flip-flops to form a sequential circuit:

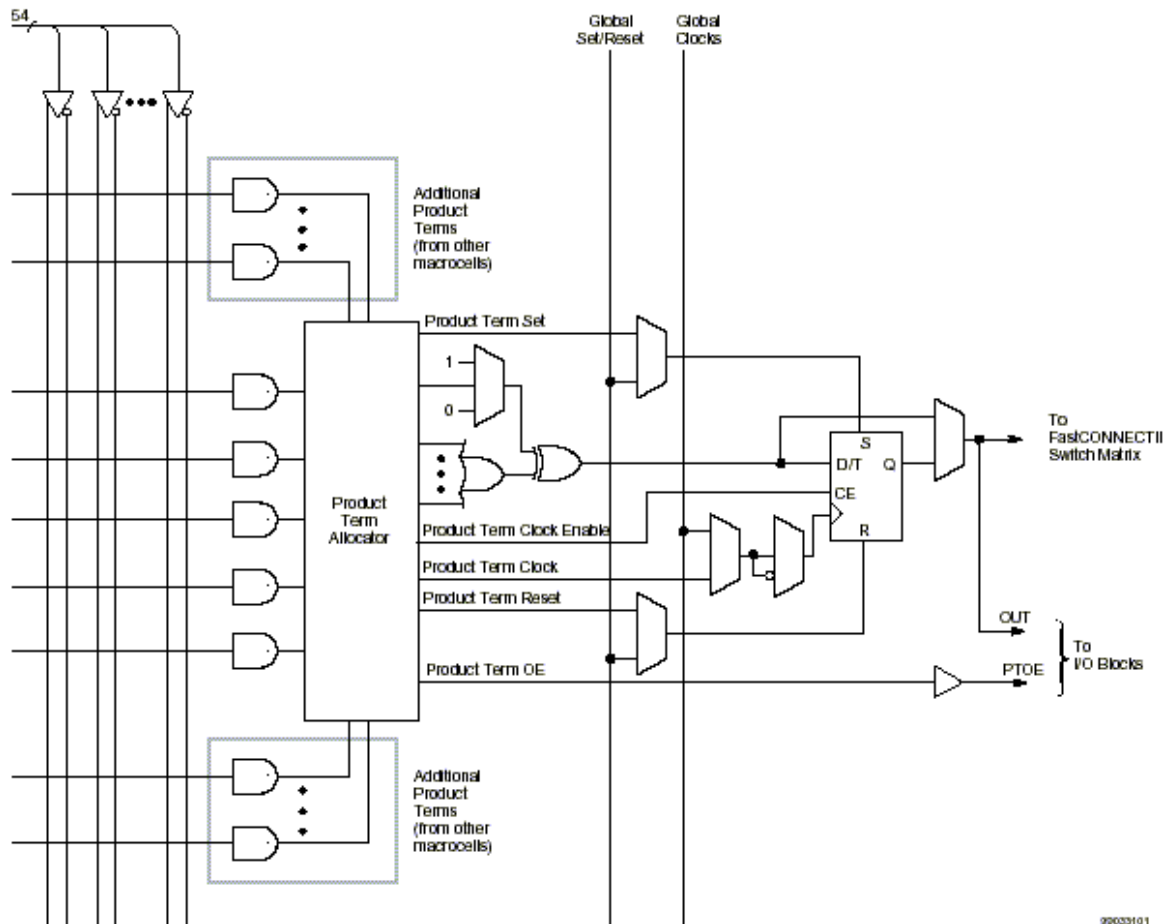
- A pair of JK flip-flops made with D or T flip-flops
- A modulo-3 counter made with two JK flip-flops
- An 8-bit shift right/shift left register with broadside load
- A 4-bit up counter made using D or T flip-flops
- A divide-by-60 counter that supplies 4-Hz, 2-Hz, and 1-Hz outputs from 60Hz input
- A 4-bit down counter made from D or T flip-flops
- A 5-bit down counter that loads a new count upon reaching 00000
- A 6-state traffic-light state machine. (Advanced with a pushbutton).
- A traffic-light decoder for the traffic-light state machine
- A 9.6 KBaud serial transmitter, loaded with data from DIP switches
- A 9.6 KBaud serial receiver, the received data appearing on the LEDs

The traffic-light experiment proves popular with students; almost everyone knows how one works. In the past, it would take quite a few ICs to implement even a simple traffic light, but now, a single, low-cost CPLD can provide virtually all the logic required by a traffic-light controller. Students can develop many of the modules that make up a traffic-light controller in independent experiments. Then, they can develop the logic for the controller and combine and test the elements they developed. In this way, they start with simple projects and work up to a complete clocked-logic controller. A simple traffic light is a special case of a state machine that progresses from state to state in an unvarying sequence.

The clock signal that triggers each state derives from the 60-Hz power-line signal or from an on-board oscillator circuit. The Lab9500board also provides an alternate time base: A PIC microcontroller with an internal 4-MHz (factory trimmed) oscillator. A short program within the PIC device generates a clock signal at a variety of frequencies from 2 Hz to as high as 76.9 KHz. (Students do not program the PIC; it comes ready to use.)

What's a Macrocell?

As already mentioned, the XC9500 family uses a PAL-style architecture. The PAL architecture is an OR output with a fixed number of AND-term inputs, or a so-called AND-OR architecture. The original PALs had a fixed polarity output, either high- or low-true. Registered PALs used the AND-OR logic as the D input of a D-type flip-flop. A pin was available for the clock input. Another pin controlled a three-state output. When this architecture was implemented in CMOS, many additional features were added. The first MOS implementations used an EPROM cell as the programmable links, requiring a windowed package. Later implementations (such as the GAL by Lattice) use EEPROM technology and are reprogrammable without the need for a windowed package. The original PALs had either combinational outputs or registered (flip-flop) outputs. The GAL has a general-purpose input/output cell called a macrocell. Under program control, this



XC9500XL Macrocell Within Function Block

cell can be configured as an input, combinational output or registered output. The output polarity is programmable. Clock sources and many other features are programmable.

A macrocell in the XC9500 family consists of an OR gate with five product terms, a flip-flop, and many programmable options such as output polarity. Some of these product terms are “stolen” for other macrocell functions (if selected), reducing the number into the OR gate. However, unused product terms from other cells can be used with cells that do not have enough product terms of their own. The OR gate is followed by a flip-flop of either the D or T type. A selector selects the output of a cell to be either the AND/OR logic, or the flip-flop. The output of a macrocell typically goes to an I/O pin on the chip and is available as an input to any AND product term. If an I/O pin is used to provide an input from the world, then the macrocell associated with the pin is not otherwise useable. Note, that J-K flip-flops are not available, because it would take two AND/OR sections per macrocell. A complete description of the XC9500XL family can be found at:

www.xilinx.com/partinfo/9500xl.pdf

Lab9500 is a trademark of EZ-Ware, Buffalo, NY.
PAL is a trademark of Applied Micro Devices